Binary Classification with DNN

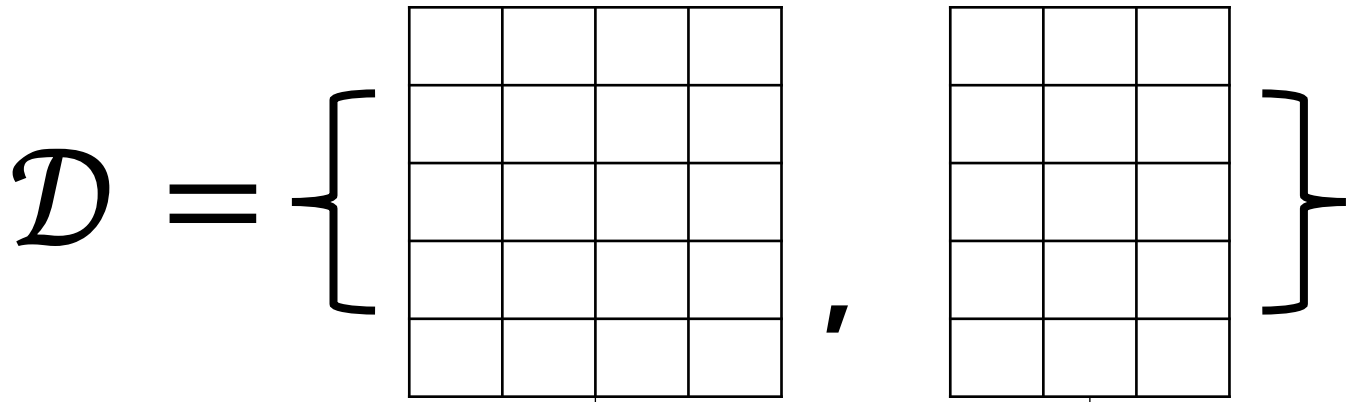
Ki Hyun Kim

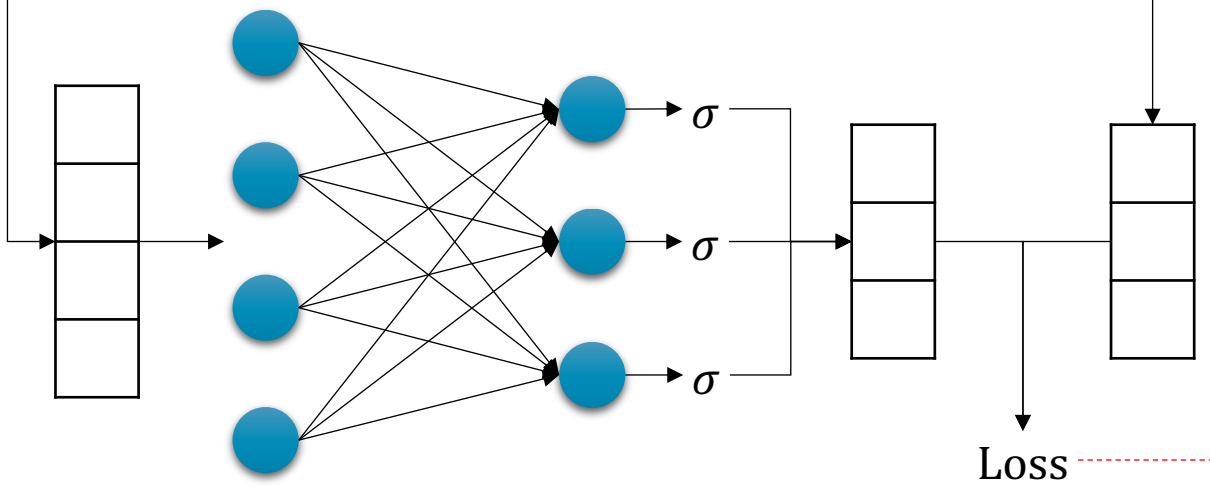
nlp.with.deep.learning@gmail.com

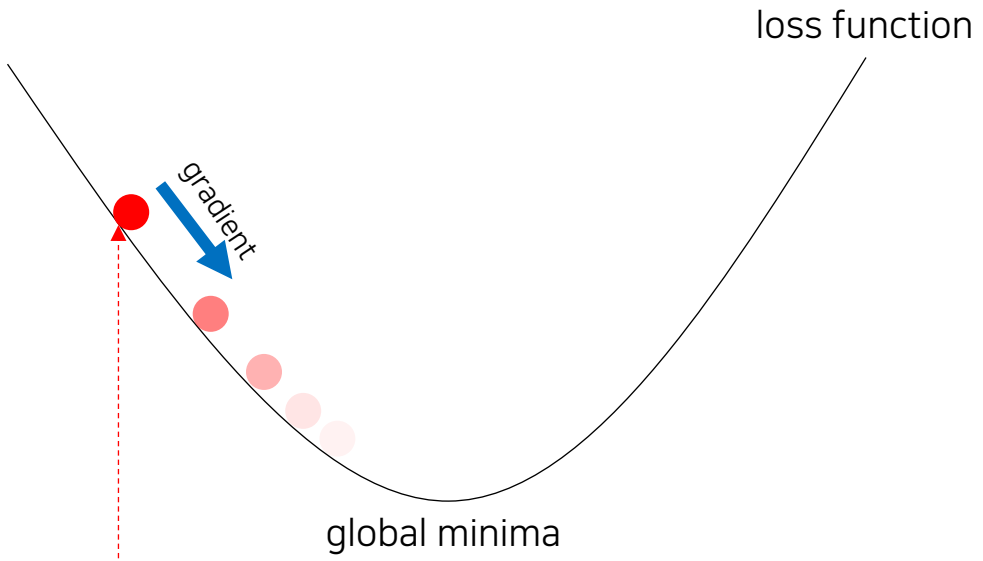
Fast campus

Review: Logistic Regression

- input_size: n
- output_size: m



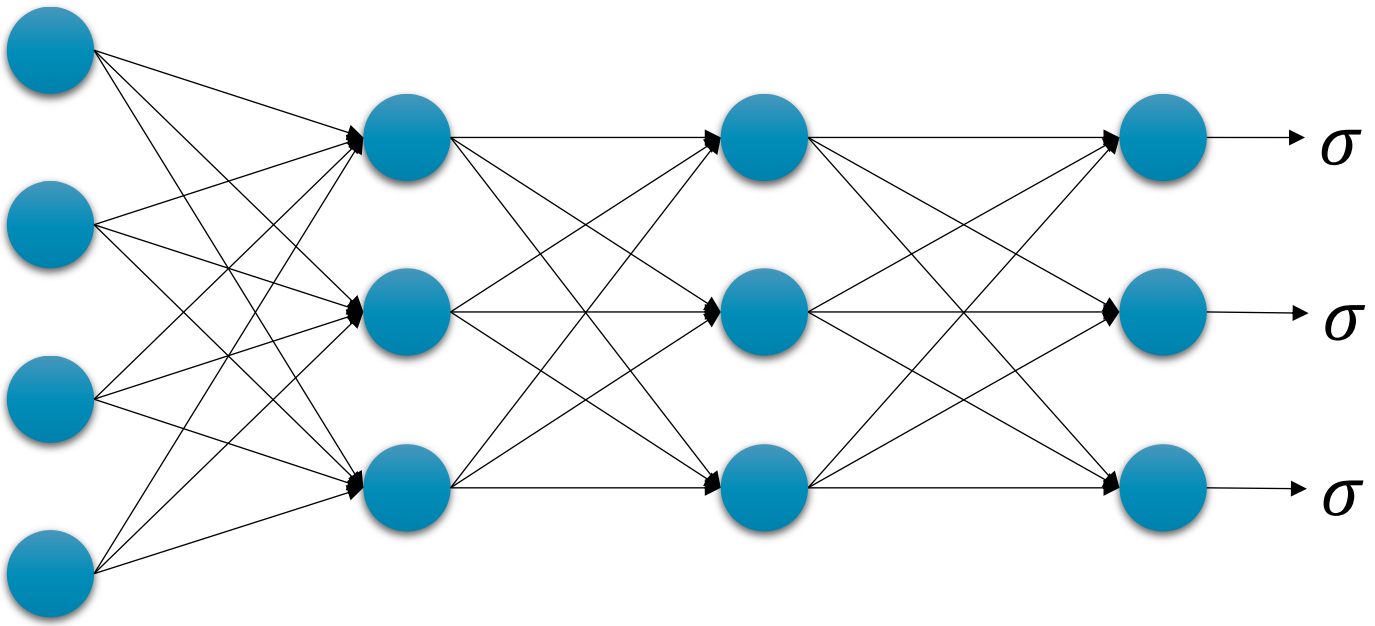




Fast campus

Architecture

- 레이어를 깊게 쌓은 이후에, 마지막 Linear Layer 이후, Sigmoid를 씌워줌

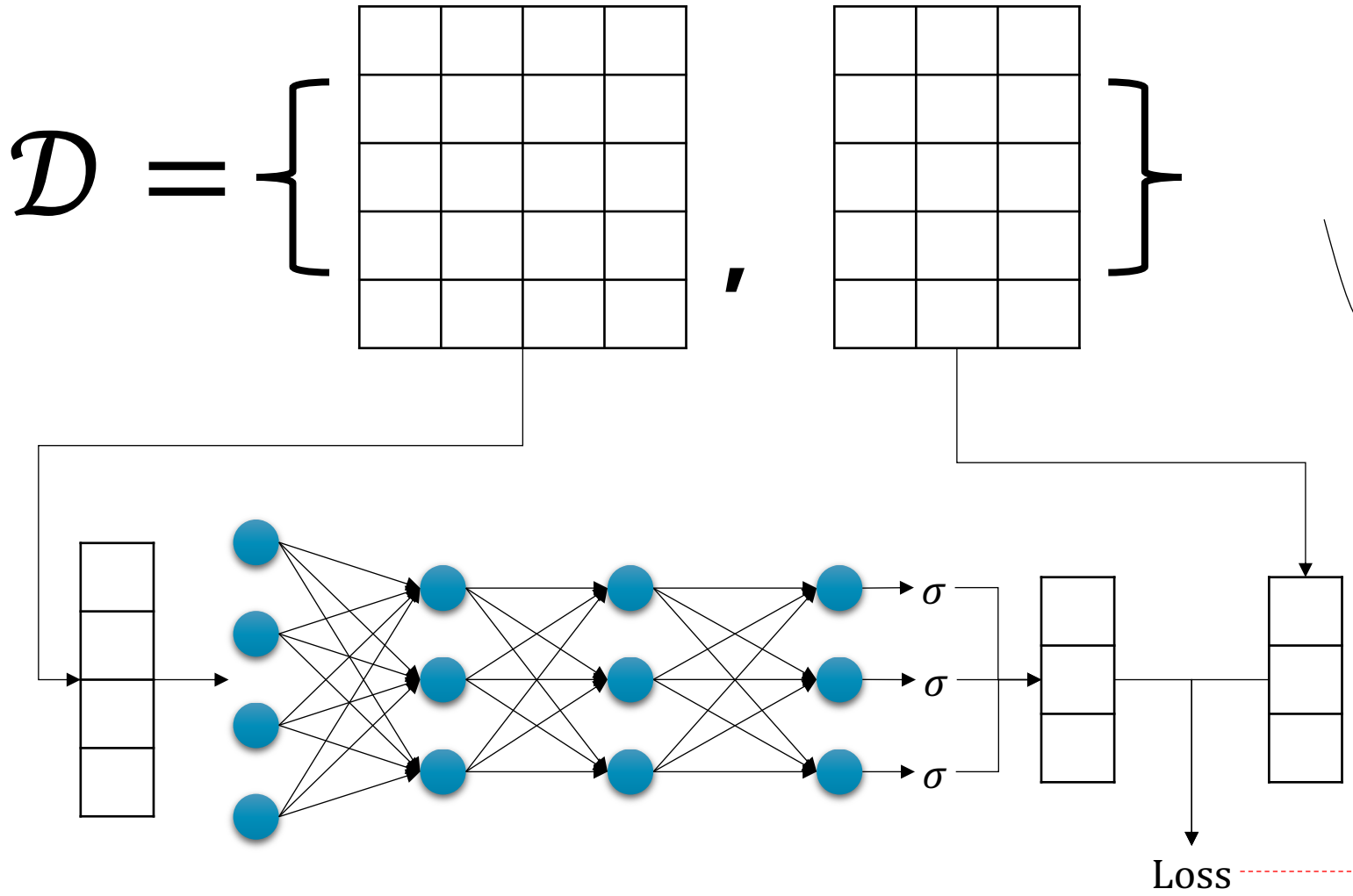


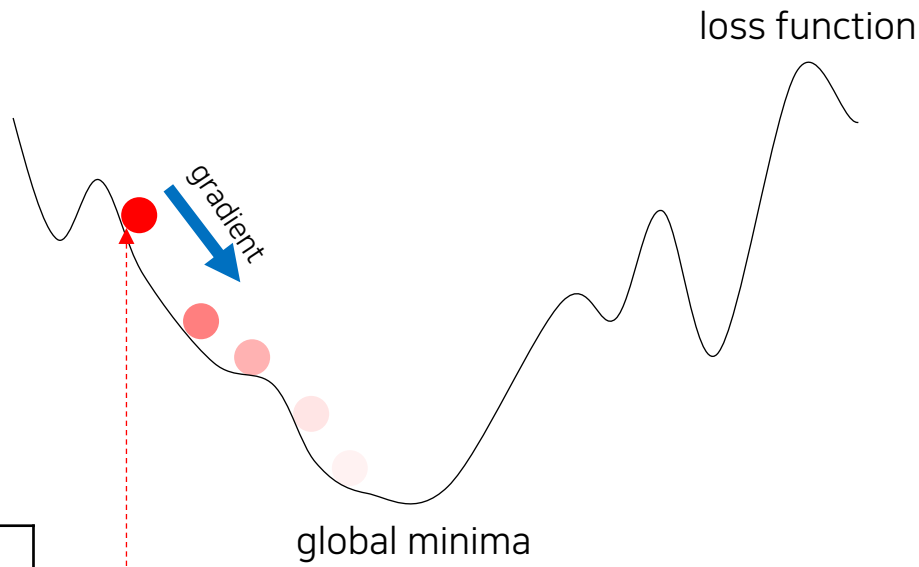
Fast campus

Binary Classification Overview

- 레이어가 깊어진 것 이외에는 Logistic Regression과 전부 똑같다.

* 앞서 코드를 그대로 활용할 수도 있다!





Fast campus

Review: Because we use Sigmoid,

- Sigmoid의 출력 값은 0에서 1

- 따라서 확률 값 $P(y|x)$ 으로 생각해볼 수 있음

$$0 \leq P(y = \text{True}|x) \leq 1$$
$$P(y = \text{True}|x) = 1 - P(y = \text{False}|x)$$

Binary Cross Entropy (BCE) Loss Function

- N개의 vector들이 주어졌을 때,

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N},$$
$$\text{where } x_{1:N} \in \mathbb{R}^{N \times n} \text{ and } y_{1:N} \in \mathbb{R}^{N \times m}.$$

$$\hat{y}_i = f_\theta(x_i)$$

$$\text{BCE}(y_{1:N}, \hat{y}_{1:N}) = -\frac{1}{N} \sum_{i=1}^{N} y_i^\top \cdot \log \hat{y}_i + (1 - y_i)^\top \cdot \log (1 - \hat{y}_i)$$

$$\hat{\theta} = \underset{\theta \in \Theta}{\text{argmin}}\, \text{BCE}(y_{1:N}, \hat{y}_{1:N})$$

Fast campus

Wrap-up

- Deep Regression과 마찬가지로, 모델을 DNN으로 교체 후, sigmoid를 마지막에 넣어준다.
  - 여전히 gradient descent 방식으로 똑같이 최적화 가능